

Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Ingeniería del Software II
Segundo cuatrimestre 2009

Primera Iteración

Matias Blanco 508/05
Diego Maximiliano Freijo Y Villoldo 4/05
Fernando Bugni 611/05
Pablo Laciana 527/05

Revisiones

Fecha	Versión	Descripción	Autor/es
27/09/09	1.0	planificación general, planificación detallada de primera iteración de elaboración	Matias Blanco, Diego Maximiliano Freijo Y Villoldo, Fernando Bugni, Pablo Laciana
29/10/09	2.0	Creación de documento, planificación general, planificación detallada de primera iteración de elaboración	Matias Blanco, Diego Maximiliano Freijo Y Villoldo, Fernando Bugni, Pablo Laciana

Índice:

Documento de planificación	4
1. Objetivo del sistema	4
2. Casos de uso.....	4
2.1. Listado de Actores	4
2.2. Listado de Casos de Uso.....	5
3. Planificación de las iteraciones	7
3.1. Fase de Incepción	7
3.2. Fase de Elaboración	8
3.2.1. Primera iteración	8
3.2.1.1. Prueba de concepto.	9
3.2.1.2. Diagrama de Gantt	11
3.2.1.3. Diagrama de Pert	13
3.2.2. Segunda iteración	15
3.3. Fase de Construcción	15
3.3.1. Primera iteración	15
3.3.2. Segunda iteración	15
3.3.3. Tercera iteración.....	15
3.4. Fase de Transición.....	16
3.5. Fechas e Hitos importantes	16
3.5.1. Fechas por etapa	17
3.5.2. Hitos y Entregables	17
4. Elaboración: Primera iteración	18
4.1. Casos de uso a implementar	18
4.1.1. Ingresando nuevo modelo Matemático	18
4.1.2. Recibiendo datos de Terminales remotas (lado Estación central)	18
4.1.3. Pidiendo información a Terminales remotas (lado Estación central)	19
4.1.4. Actualizando datos.....	19
4.1.5. Pidiendo información (lado Terminales remotas)	19
4.1.6. Enviando datos a Estación central (lado Terminales remotas) ...	20

4.2. Atributos de calidad y escenarios.....	20
4.2.1. Confiabilidad:	20
4.2.2. Disponibilidad:	21
4.2.3. Performance:	22
4.2.4. Seguridad:.....	23
4.2.5. Modificabilidad:	23
4.2.5. Posibles cambios y supuestos fundamentales.....	24
4.2.5.1. Supuestos fundamentales.....	24
4.2.5.2. Posibles cambios	25
4.3. Plan de Riesgos.....	25
4.3.1. Disponibilidad del servicio de SMS.....	25
4.3.2. Consumo de energia de las TR.....	25
4.3.3. No disponibilidad de TR	26
4.3.4. Datos erroneos a partir de los modelos predictivos	26
4.3.5. Caída del servicio web para alertar a la poblacion.....	26
4.3.6. Stoplight Chart:	27
Documento de Arquitectura	28
5. Organización del documento.....	28
5.1. Partes del documento	28
5.2. Glosario	28
6. Visión general de la arquitectura.....	29
6.1. Consideraciones Generales	29
7. Diagrama de arquitectura	30
7.1. Diagrama	30
7.2. Detalles del diagrama	32
7.3. Dinámica general del sistema.....	32
8. Arquitectura y atributos de calidad	34
8.1. Seguridad	34
8.2. Disponibilidad	34
8.3. Performance	35
8.4. Confiabilidad.....	35
8.5. Modificabilidad	36
Seguimiento y avance	37
12. Informe de seguimiento y avance.....	37
13. Prueba de concepto.....	40
13.1. Descripción.....	40
13.2. Tecnologías utilizadas	40

Documento de planificación

1. Objetivo del sistema

El presente documento intentará documentar todas las decisiones de que se tomen en relación con el sistema de mediciones telemétricas pensado por el Gobierno. Se han llevado a cabo reuniones con los actores que forman parte del proyecto y hemos extraído información de ellas para poder llevar a cabo, mediante el desarrollo de sistemas, la satisfacción de los requerimientos de los usuarios.

El objetivo del proyecto es poder lograr una red que sea capaz, gracias a sus mediciones, de recolectar la mayor cantidad de datos posible con la mayor confiabilidad para que permita anticiparse al desarrollo de fenómenos meteorológicos. También la capacidad de la definición y modificación de modelos matemáticos capaces de determinar si un determinado fenómeno ocurre durante los próximos días.

Para esto nos hemos basado en Unified Process y hemos utilizado algunos de los templates que define este método, hemos definido las fases del proyecto y cuales son sus iteraciones. También debimos definir los hitos y entregables que van a acompañar el desarrollo de todo el proyecto para, por último, estimar la fecha de finalización del mismo.

Es importante aclarar que, debido a que la cátedra es la que va a definir cuál será la segunda iteración en la fase de construcción, todas las estimaciones pueden verse afectadas dependiendo de qué sea lo que hay que implementar en la segunda etapa del trabajo práctico.

2. Casos de uso

En esta primer sección del documento se hace un listado exhaustivo de los casos de uso de alto nivel que se pudieron extraer a partir de los requerimientos elicitados en las reuniones mantenidas con los diferentes stakeholders.

Si bien estas reuniones han ocurrido en el contexto de la fase de inepción y los casos de uso son un primer resultado de la fase de elaboración, se ha decidido presentar este listado en primer lugar y aislado del resto dado que la metodología aplicada está fuertemente guiada por los casos de uso y se los quiere dotar de alta visibilidad.

2.1. Listado de Actores

Este es un listado de los actores que interactúan con el sistema en los casos de uso que luego especificaremos:

1. **Administrador:** Usuario con permisos suficientes para ingresar nuevos modelos matemáticos, alarmas y altas bajas y modificaciones de TRs.
2. **Biggest Satelit:** Cualquier interfaz posible para poder obtener datos de esta empresa

3. **Sistema de pagos Ministerio:** Cualquier interfaz posible para poder informar el consumo del sistema
4. **Visualizador de información:** Cualquier usuario web con la posibilidad de ver los datos meteorológicos.
5. **Sistema eólico:** Cualquier sistema eólico capaz que enviar datos a la estación central por web services
6. **Operador:** Cualquier operario capaz de consultar el estado de la red de TRs
7. **Sensor:** Cualquier dispositivo sensor que cumpla con el protocolo planteado en la arquitectura de la TRs. Debe cumplir con la función de ir actualizando los datos registrados.
8. **Reloj:** Cualquier dispositivo confiable del cual pueda obtenerse la hora exacta.

Aclaración: decidimos tomar a Sensor como un Actor del Sistema ya que es un componente que se agrega para sensar los datos. Posee un protocolo de comunicación que utiliza el sistema. El sistema en sí, no define un Sensor sino que lo utiliza como un Actor para poder obtener datos meteorológicos. Lo mismo pasa con Reloj.

2.2. Listado de Casos de Uso

1. **Ingresando configuración de alarmas**
Descripción: Se debe permitir a cualquiera que sea Administrador la posibilidad de cambiar la configuración de las Alarmas
Actor principal: Administrador
Dependencias:
2. **Ingresando nuevo modelo Matemático**
Descripción: Se debe permitir a cualquiera que sea Administrador la posibilidad de ingresar un nuevo modelo Matemático
Actor principal: Administrador
Dependencias:
3. **Ingresando intervalo de configuración**
Descripción: Se debe permitir a cualquiera que sea Administrador la posibilidad de ingresar la configuración de los intervalos que utilizan los sensores para reportar a las TRs.
Actor principal: Administrador
Dependencias:
4. **Alta baja y modificación de Terminales remotas**
Descripción: Se debe permitir a cualquiera que sea Administrador la posibilidad de dar de baja, de alta o modificar una determinada TR
Actor principal: Administrador
Dependencias: Terminal remota involucrada
5. **Actualizando información en Terminales remotas**
Descripción: Luego de modificar algún dato sobre la TR se debe dar la posibilidad de enviarle la configuración nueva
Actor principal: Administrador
Dependencias: disponibilidad de gateway/s SMS
6. **Recibiendo datos de Terminales remotas (lado Estación central)**

Descripción: Se debe permitir enviar información de cada TR sobre los datos que recolecta

Actor principal: Administrador

Dependencias: disponibilidad de gateway/s SMS

7. Pidiendo información a Terminales remotas (lado Estación central)

Descripción: Se debe permitir a la estación central tener la capacidad de que pueda pedir la información de las Terminales remotas via SMS

Actor principal: Administrador

Dependencias: disponibilidad de gateway/s SMS

8. Obteniendo datos de Biggest Satelit

Descripción: Se debe tener la capacidad de poder obtener los datos de esta empresa para poder monitorear el consumo

Actor principal: Biggest Satelit

Dependencias: Conexión con Biggest Satelit

9. Informando consumo

Descripción: Se debe tener la capacidad de que la Estación central pueda informar el consumo al Sistema Pagos del Ministerio

Actor principal: Sistema Pagos del Ministerio

Dependencias: Conexión con Sistema Pagos del Ministerio

10. Consultando datos meteorológicos

Descripción: Se debe tener la capacidad de que la Estación central pueda informar sobre los datos obtenidos

Actor principal: Visualizador de informacion

Dependencias: disponibilidad de web server/s, servicio de Google Maps disponible

11. Enviando datos a Estación central por web services

Descripción: Se debe tener la capacidad de poder enviar datos desde el Sistema eólico hacia la Estación central por web services

Actor principal: Sistema eólico

Dependencias: disponibilidad de Web Services

12. Enviando alarma por email

Descripción: Se debe tener la capacidad de que la Estación central pueda enviar las alarmas por email

Actor principal: Sensor

Dependencias: Servidor de correo (SMTP)

13. Consultando estado de la red

Descripción: Se debe tener la posibilidad de un Operador pueda consultar el estado de la red a través de la Estación central

Actor principal: Operador

Dependencias:

14. Informar estado de batería y radiación solar

Descripción: Se debe tener la capacidad de que el Sistema administrador de energía tenga la capacidad de informar el estado de la batería y de la radiación solar

Actor principal: Sistema administrador de energía

Dependencias:

15. Actualizando datos de Sensor

Descripción: Se debe tener la capacidad de que el sensor actualice los datos de sus mediciones

Actor principal: Sensor

Dependencias: Terminal remota

16. Actualizando estado de sensor

Descripción: Se debe tener la posibilidad de que la Terminal remota pueda actualizar el estado del sensor

Actor principal: Sensor

Dependencias: Terminal remota

17. Actualizando configuración de sensor

Descripción: Se debe tener la capacidad de desde la Estación central enviarle una señal a la TR para poder actualizar la nueva configuración de un sensor determinado

Actor principal: Administrador

Dependencias: Sensor

18. Pidiendo información (lado Terminales remotas)

Descripción: Se debe tener la capacidad de poder efectuar los pedidos de información de la Estación central. Esta información puede ser el estado de la TR o sus datos sensados

Actor principal: Administrador

Dependencias: disponibilidad de gateway/s SMS

19. Enviando datos a Estación central (lado Terminales remotas)

Descripción: Se debe permitir enviar información de cada TR sobre los datos que recolecta

Actor principal: Reloj

Dependencias: disponibilidad de gateway/s SMS

3. Planificación de las iteraciones

3.1. Fase de Incepción

En la iteración inicial, correspondiente a la fase de incepción del proyecto, se realizó la elicitación de requerimientos a partir de la información provista por el cliente. Se identificaron los principales componentes del sistema con sus responsabilidades. A partir del diálogo con los principales stakeholders se priorizaron los atributos de calidad (luego de un Quality Attribute Workshop organizado a tal fin). Se listaron los principales casos de uso del sistema teniendo en cuenta sus dependencias y riesgos para poder realizar una planificación adecuada del resto del proyecto.

Tanto los casos de uso como la priorización de atributos de calidad fue incluida en secciones independientes al principio del documento para otorgarles mayor visibilidad.

En conclusión, las tareas realizadas para la iteración correspondiente a la fase de incepción fueron:

- Relevamiento inicial: se comenzaron a delinear requerimientos en reuniones preliminares con el cliente.

- QAW: se realizó un Quality Attribute Workshop para traer a la superficie los atributos de calidad requeridos. Del workshop surgieron también más datos sobre la definición de los requerimientos.
- Refinación de requerimientos y documentación en casos de uso: luego de las reuniones mencionadas, se refinaron los requerimientos elicitados y se los plasmó en un listado inicial de casos de uso.

3.2. Fase de Elaboración

Los objetivos de la fase de elaboración son los de analizar el dominio del problema, verificar si los requerimientos detallados están bien especificados, verificar si la arquitectura satisface los requerimientos y poder realizar una estimación más precisa que en el primer momento del proyecto.

3.2.1. Primera iteración

La primera iteración va a estar basada en interiorizarnos en el problema en cuestión. Gracias a ello vamos a poder definir requerimientos funcionales y restricciones. También vamos a verificar que pueda cumplirse los requisitos de calidad con los requerimientos que tenemos.

Teniendo en cuenta esto, los casos de uso incluídos en esta iteración son:

- 2 - Ingresando nuevo modelo Matemático
- 6 - Recibiendo datos de Terminales remotas (lado Estación central)
- 7 - Pidiendo información a Terminales remotas (lado Estación central)
- 15 - Actualizando datos de Sensor
- 18 - Pidiendo información (lado Terminales remotas)
- 19 - Enviando datos a Estación central (lado Terminales remotas)

Tareas:

A continuación se muestran las tareas identificadas y su respectiva estimación. El método utilizado para la estimación es el llamado Poker Planning. El método consiste en lo siguiente: cada integrante del equipo se le da una serie de cartas con los numeros 1, 2, 4, 8 y 16 que representan la cantidad de horas que va a llevar a cabo una tarea. Se expone una tarea y se da una breve descripción de la misma. Luego cada integrante elige la cantidad de horas que llevará dicha tarea. Si todos coinciden, se elige esa estimación para esa tarea. Sino cada integrante del equipo expone sus razones de su estimación. Se repite el procedimiento un par de veces, y si no hay acuerdo, se elige a uno de los integrantes para definir la cuestión. Para cada tarea, la persona que define es determinada de acuerdo a su experiencia, conocimiento y/o aptitudes relacionadas a la misma. Cabe aclarar que un día equivale aproximadamente a 4 horas.

Utilizamos este método ya que utiliza fuertemente al equipo de trabajo y se llega a un consenso para cada persona involucrada en el equipo.

Las tareas identificadas para esta iteración son:

- * Elaborar plan de proyecto
 - Crear un documento inicial con estimaciones tentativas. (2 días)

- Ajustar el documento al final de la iteración. (1 día)
- * Confeccionar lista de Casos de Uso
 - Análisis del modelo inicial de Casos de Uso. (1 día)
 - Documentación de casos de uso a alto nivel. (2 días)
- * Ejecutar análisis de riesgos
 - Identificar y analizar fuentes de riesgos. (1 día)
 - Crear un plan de riesgos para mitigarlos. (1 día)
- * Identificar atributos de calidad
 - Quality Attribute Workshop (1 día)
 - Documentación de los escenarios identificados en el QAW (2 días)
- * Definir arquitectura del sistema (requiere: atributos de calidad)
 - Análisis de tecnologías. (1 día)
 - Definición de estilo arquitectónico. (1/2 día)
 - Documentar un modelo de arquitectura. (3 días)
 - Evaluación de la arquitectura propuesta. (2 días)
- * Elaborar Informe de avance
 - Crear un documento que describa el avance del proyecto (1/2 día)
- * Desarrollar prototipo (ver detalle a continuación)

3.2.1.1. Prueba de concepto.

Para poder cumplir con un atributo de calidad clave, debemos realizar en esta iteración el desarrollo de un prototipo que pueda lograr una transmisión segura y con completitud de datos censados desde varias TRs simuladas y de la configuración de un modelo matemático simple desde la Estación Central que utilice los datos enviados por las TRs. Para alcanzar este objetivo y para poner a prueba la arquitectura propuesta, desarrollaremos una pequeña porción de esta última que pueda cubrir la prueba de concepto.

Las tareas que involucran dicho prototipo son:

Preparación de entorno de desarrollo: 1 día

Esta tarea incluye crear el proyecto en el IDE correspondiente, obtención y referenciación de librerías a utilizar y el agregado de todos estos artefactos al repositorio.

Modulo de ingreso nuevo modelo Matemático: 4 días

- Diseño del módulo: diseñar un módulo lo suficientemente sofisticado como para poder cargar cualquier modelo Matemático y a partir de ello lograr que haga el calculo del modelo
- Implementación del módulo
- Diseño de casos de test para el módulo

- Implementación de casos de test para el módulo
- Corridas de tests, documentación de resultados y generación de tareas de corrección
- Corrección de bugs
- Corridas definitiva de tests para asegurar la correcta corrección de bugs

Modulo Interfaz de ingreso nuevo modelo Matemático: 4 días

- Diseño del módulo: creación de una interfaz intuitiva para lograr hacer facil la carga de modelos Matemáticos
- Implementación del módulo
- Diseño de casos de test para el módulo
- Implementación de casos de test para el módulo
- Corridas de tests, documentación de resultados y generación de tareas de corrección
- Corrección de bugs
- Corridas definitiva de tests para asegurar la correcta corrección de bugs

Investigación de protocolo seguro de comunicación: 2 días

Averiguar algoritmos y protocolos de comunicación de modo seguro

Modulo de recepción de datos de Estación Central: 5 días

- Diseño del módulo: diseñar un modulo del lado de la Estación Central capaz de recibir los datos de las diferentes TRs
- Implementación del módulo
- Diseño de casos de test para el módulo
- Implementación de casos de test para el módulo
- Corridas de tests, documentación de resultados y generación de tareas de corrección
- Corrección de bugs
- Corridas definitiva de tests para asegurar la correcta corrección de bugs

Modulo de pedido de información a Terminales remotas (lado Estación Central): 5 días

- Diseño del módulo: diseñar un modulo del lado de la Estación Central con la función de pedir información a cada TRs
- Implementación del módulo
- Diseño de casos de test para el módulo
- Implementación de casos de test para el módulo
- Corridas de tests, documentación de resultados y generación de tareas de corrección
- Corrección de bugs
- Corridas definitiva de tests para asegurar la correcta corrección de bugs

Modulo de actualización de datos de los sensores: 3 días

- Diseño del módulo: diseñar un modulo con la función de actualizar los datos de los sensores para tener la mejor información.
- Implementación del módulo
- Diseño de casos de test para el módulo
- Implementación de casos de test para el módulo

- Corridas de tests, documentación de resultados y generación de tareas de corrección
- Corrección de bugs
- Corridas definitiva de tests para asegurar la correcta corrección de bugs

Modulo de pedido de información (lado Terminales remotas): 5 días

- Diseño del módulo: diseñar un modulo capaz de pedirle información a la Estación Central para actualizar su estado
- Implementación del módulo
- Diseño de casos de test para el módulo
- Implementación de casos de test para el módulo
- Corridas de tests, documentación de resultados y generación de tareas de corrección
- Corrección de bugs
- Corridas definitiva de tests para asegurar la correcta corrección de bugs

Modulo de envio de datos a Estación Central (lado Terminales remotas): 5 días

- Diseño del módulo: diseñar un modulo con la función de enviarle los datos sensados a la Estacion Central
- Implementación del módulo
- Diseño de casos de test para el módulo
- Implementación de casos de test para el módulo
- Corridas de tests, documentación de resultados y generación de tareas de corrección
- Corrección de bugs
- Corridas definitiva de tests para asegurar la correcta corrección de bugs

Correcciones a la planificación: 2 días

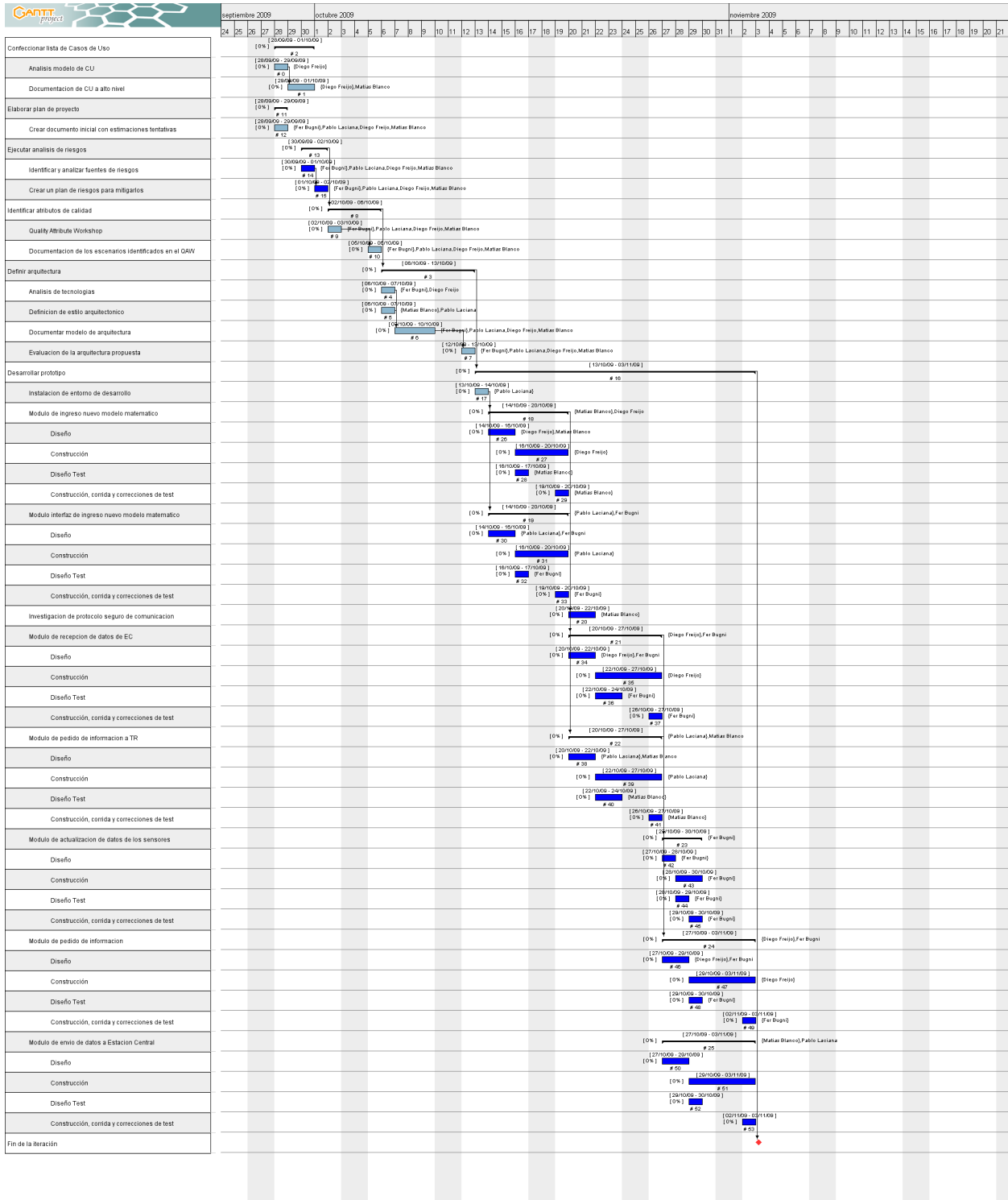
Relevar el estado de avance de la prueba de concepto, adicionar al documento todo conocimiento adquirido, introducir correcciones al calendario o al alcance de cada iteración de acuerdo al feedback de la iteración.

Para realizar las estimaciones, utilizamos la experiencia de un integrante de este proyecto que posee experiencia en el desarrollo de aplicaciones de comunicaciones con un alto grado de performance

3.2.1.2. Diagrama de Gantt

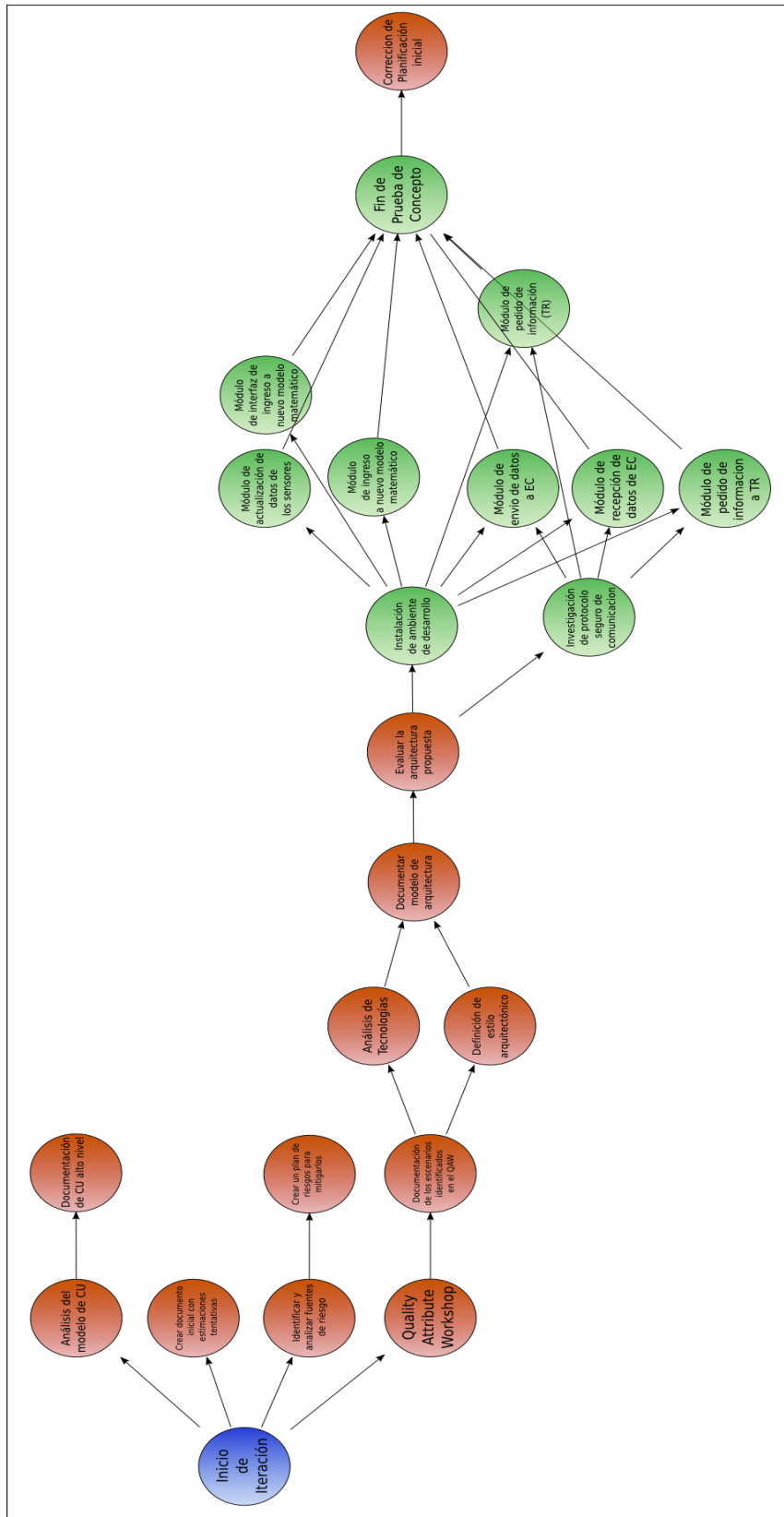
Con el fin de documentar las tareas, su duración, dependencias entre sí y la asignación de recursos a las mismas de la forma más clara posible, se provee un diagrama de Gantt.

Cabe aclarar que en varias ocasiones consideramos necesario dividirnos en pequeños grupos de dos personas con el objetivo de enfocarnos en una tarea determinada mientras que el resto del equipo de desarrollo se encarga de otras. Sobre las tareas de mayor complejidad decidimos que la participación de todo el equipo era fundamental.



3.2.1.3. Diagrama de Pert

Para lograr una mejor comprensión de las relaciones entre las tareas y cuál es su correlato, hemos diseñado el siguiente grafo al estilo Pert en donde se pueden observar qué tareas preceden a qué otras.



3.2.2. Segunda iteración

Esta fase se basa principalmente en la realización de una arquitectura completa para el sistema que necesitamos. Basandonos con la arquitectura usada en la Prueba de Concepto, vamos a terminar de delimitar la arquitectura definitiva para cumplir con los requerimientos de Calidad.

Esta iteración involucra los siguientes casos de uso:

- 13 - Consultando estado de la red
- 14 - Informar estado de batería y radiación solar

3.3. Fase de Construcción

En la presente fase se especificará, desarrollará y testeará el producto de manera incremental. La elección de los casos de uso se basó en el riesgo que este puede causar para fracasar el proyecto. Con la arquitectura definida en su totalidad y la prueba de Concepto realizada, nos enfocamos primero en la totalidad de la funcionalidad sobre la arquitectura planteada y la puesta a punto físicamente de la misma.

3.3.1. Primera iteración

Esta primera iteración se basa en completar las funcionalidades críticas del sistema que tienen estrecha conexión con la arquitectura. Cabe aclarar que hacia la finalización de esta iteración estaría el core del sistema estaría funcionando.

Estos casos de uso son:

- 4 - Alta baja y modificación de Terminales remotas
- 5 - Actualizando información en Terminales remotas
- 11 - Enviando datos a Estación central por web services

3.3.2. Segunda iteración

En esta segunda iteración debemos abordar los casos de uso que tienen que ver con los clientes directos. Esto es Biggest Satelit y la Gobernación

- 8 - Obteniendo datos de Biggest Satelit
- 9 - Informando consumo
- 16 - Actualizando estado de sensor
- 17 - Actualizando configuración de sensor

3.3.3. Tercera iteración

En la tercera y ultima iteración nos dedicaremos a los casos de uso que son funcionalidades de datos ya existentes en el sistema. Al tener ya creado el sistema y sus datos, estas funcionalidades son solo una forma de mostrar esos datos o alguna configuración que soporta la arquitectura.

Los casos de uso son:

- 1 - Ingresando configuración de alarmas
- 3 - Ingresando intervalo de configuración
- 10 - Consultando datos meteorológicos
- 12 - Enviando alarma por email

3.4. Fase de Transición

En el presente documento especificaremos cómo se desarrollará la fase de transición del proyecto. El propósito de la fase de transición es acercar el software desarrollado a los usuarios. Una vez que los usuarios toman contacto con el sistema, pueden generarse sucesivas etapas de corrección de problemas y generar nuevas versiones.

Debemos entrar en la fase de Transición cuando estemos en la etapa de Desarrollo y el proyecto este en condiciones de acercarse al cliente, debemos tener un nivel de calidad aceptable y documentación pertinente como para que el cliente pueda tener un resultado aceptable.

Las etapas de la fase de transición serán las siguientes:

- Testing del producto desarrollado, de manera de validar el sistema de acuerdo a las expectativas de los usuarios.
- Entrenamiento de los usuarios en el uso del sistema.
- Detección de requerimientos de cambios en el sistema, producto del uso por parte de los usuarios finales.
- Modificaciones en base a estos requerimientos de cambio, especificación, diseño, implementación, testing de los mismos y nuevo release.

El objetivo en esta fase de Transición es conseguir que los stakeholders estén al tanto de que se ha llegado a un estado del sistema en donde las líneas de deployment son consistentes con el criterio al que se había llegado al comienzo del proyecto. Para ello es importante valorar las líneas de deployment contra la visión y el criterio de aceptación para el producto así como también obtener beneficios para el sistema en actividades de tuning, mejoras en la performance y la usabilidad de la interfaz con el usuario final.

3.5. Fechas e Hitos importantes

Decidimos hacer un esquema general de las etapas del proyecto especificando las iteraciones, fechas de los hitos más representativos y fecha de fin del proyecto. Este documento fue elaborado en una fecha temprana del proyecto por lo que las fechas de los hitos son representativas y pueden modificarse a lo largo de las diferentes etapas del proyecto.

Iremos modificando este documento mientras realizamos algunas de estas etapas, por lo que el corrector del trabajo solo verá el documento terminado.

Vale destacar que estas fechas son aproximadas y no implican un rígido orden de terminación y comienzo de la próxima etapa de cada uno de los pasos del esquema.

3.5.1. Fechas por etapa

Etapa	Fecha de Comienzo	Fecha de Finalización
Fase de Incepción: Reuniones con StakeHolders	03/09/09	16/09/09
Identificación de requerimientos Fase de Incepción: Definición de Atributos de Calidad: (WorkShop)	17/09/09	24/09/09
Fase de Incepción: Especificación de Casos de Uso:	25/09/09	27/09/09
Fase de Elaboración: Primera Iteración	28/09/09	02/11/09
Fase de Elaboración: Segunda Iteración	03/11/09	24/11/09
Fase de Construcción: Primera Iteración	25/11/09	20/12/09
Fase de Construcción: Segunda Iteración	21/12/09	22/01/10
Fase de Construcción: Tercera Iteración	23/01/10	25/02/10
Fase de Transición:	26/02/10	22/03/10

Teniendo las fechas aproximadas de comienzo y finalización de cada etapa, también podemos estimar cuales serán las fechas de los hitos o entregables más importantes en la implementación del sistema.

3.5.2. Hitos y Entregables

Hito o Entregable	Fecha
Plan de Proyecto	29/09/09
Documento con los Casos de Uso del sistema	01/10/09
Plan de Riesgos	02/10/09
Documento con atributos de Calidad.	06/10/09
Diseño de la Arquitectura	13/10/09
Prueba de Concepto - Fase de Elaboración: Primera Iteración	03/11/09
Finalización Fase de Construcción: primera iteración	20/12/09

Finalización Fase de Construcción: segunda iteración	22/01/10
Finalización Fase de Construcción: tercera iteración	25/02/10
Finalización Fase de Transición	22/03/10

Acerca de cuáles son los entregables que corresponden a la segunda etapa del trabajo práctico, hemos agregado algunos con fechas aproximadas que hemos estimado, pero que no conocemos con certeza ya que no poseemos el enunciado de esta etapa del trabajo.

4. Elaboración: Primera iteración

4.1. Casos de uso a implementar

En esta sección se detallaron los casos de uso elegidos para esta iteración.

4.1.1. Ingresando nuevo modelo Matemático

Caso de Uso: 2 - Ingresando nuevo modelo Matemático	
Actor Principal: Administrador	
Pre-Condicion: El administrador autenticado en el sistema	
Post-Condicion: El nuevo modelo matematico queda registrado en el sistema para ser utilizado en el procesamiento de la informacion.	
Curso Normal	Curso Alternativo
1. El administrador obtiene el formulario para el ingreso de la nueva formula.	
2. El administrador completa el formulario con la formula y las opciones correspondientes	
3. El sistema guarda la formula en la base de datos.	

4.1.2. Recibiendo datos de Terminales remotas (lado Estación central)

Caso de Uso: 6 - Recibiendo datos de Terminales remotas (lado Estación central)	
Actor Principal: Administrador	
Pre-Condicion: Disponibilidad del sistema de SMS.	
Post-Condicion: La informacion recibida se almacena en la base de datos de condiciones ambientales.	
Curso Normal	Curso Alternativo
1. La EC recibe los datos de la TR por uno o mas SMS.	
2. La EC junta la informacion, la descripta y la decodifica.	2. En caso de haberse detectado una modificacion tanto en la desenscripcion como

	en la decodificación, se informa al Administrador lo sucedido.
3. La EC guarda la información recibida en la base de datos para el posterior procesamiento.	

4.1.3. Pidiendo información a Terminales remotas (lado Estación central)

Caso de Uso: 7 - Pidiendo información a Terminales remotas (lado Estación central)	
Actor Principal : Administrador	
Pre-Condición: Disponibilidad del sistema de SMS	
Post-Condición: La terminal remota queda notificada del pedido de información de disponibilidad por parte de la Estación Central.	
Curso Normal	Curso Alternativo
1. La EC prepara un mensaje pidiendo la información de disponibilidad de una TR específica.	
2. La EC encodea y encripta el mensaje para la estación, para evitar una modificación en la transmisión.	
3. La EC envía el mensaje a la TR correspondiente y pone en espera el estado del pedido de información a la TR. Extiende "18 - Pidiendo información (lado Terminales remotas)"	
4. La EC recibe el mensaje de la TR informado que la misma está disponible y funcionando.	4. En caso de no recibir respuesta en el plazo de 1 minuto, envía una alarma al Administrador informando que la TR está caída.

4.1.4. Actualizando datos

Caso de Uso: 15 - Actualizando datos	
Actor Principal: Sensor	
Pre-Condición: Terminal Remota funcionando	
Post-Condición: Los nuevos datos del sensor son almacenados en la TR.	
Curso Normal	Curso Alternativo
1. El sensor envía por la conexión existente entre el y la TR la nueva información sensada.	
2. La TR guarda esa información para una posterior petición de la EC.	

4.1.5. Pidiendo información (lado Terminales remotas)

Caso de Uso: 18 - Pidiendo información (lado Terminales remotas)
--

Actor Principal: Administrador	
Pre-Condicion: Disponibilidad del sistema SMS. TR funcionando.	
Post-Condicion: La TR genera un mensaje para luego enviarlo a la EC.	
Curso Normal	Curso Alternativo
1. La TR recibe el mensaje que solicita el envío de la nueva informacion de disponibilidad de la TR.	
2. La TR envia el mensaje de disponibilidad OK.	

4.1.6. Enviando datos a Estación central (lado Terminales remotas)

Caso de Uso: 19 - Enviando datos a Estación central (lado Terminales remotas)	
Actor Principal: Reloj	
Pre-Condicion: Disponibilidad del sistema SMS. TR funcionando.	
Post-Condicion: El mensaje con la informacion de los sensores, encriptado y encodeado, se envia a la EC.	
Curso Normal	Curso Alternativo
1. La TR recibe el aviso del reloj avisando que paso el tiempo para enviar la informacion a la EC.	
2. La TR arma el mensaje con la informacion recoletada por los sensores.	
3. La TR encodea y encripta el mensaje para la transmision segura de la informacion.	
4. La TR envia por SMS el mensaje generado en el paso 3 a la EC.	

4.2. Atributos de calidad y escenarios

Se han identificado atributos de calidad necesarios para la aceptación del producto. En la siguiente lista se especifican escenarios para los atributos de calidad más importantes. La especificación se ha basado en el resultado del Quality Attributes Workshop realizado en clase junto a la cátedra y por consideraciones particulares del equipo de desarrollo. Se espera que de los siguientes escenarios se puedan desprender tácticas de arquitectura para poder sobrellevar estos requerimientos.

4.2.1. Confiabilidad:

La información recolectada de las diferentes fuentes no puede tener errores. Se debe tener alta precisión en los resultados que entregue el sistema al usuario. Es necesario evitar alarmar a la población con pronósticos erróneos, y no sorprenderse con fenómenos meteorológicos no previstos.

Escenario 1:

Se deben utilizar datos históricos para validar las mediciones nuevas.

- Fuente: Interna
- Estimulo: Ingresa un nuevo dato incorrecto
- Artefacto: Modulo de recepción de datos
- Entorno: Normal
- Respuesta: El dato es enviado a los expertos correspondientes para que lo validen o rechazen
- Medida: Los datos erroneos son identificados en un 99,9% de los casos

Escenario 2:

Se debe asegurar el orden de los datos a medida que van llegando de las terminales.

- Fuente: Terminal Remota
- Estimulo: El arribo de un mensaje
- Artefacto: Modulo de recepción de datos
- Entorno: Normal
- Respuesta: El numero de serie del mensaje debe ser posterior al último recibido desde la misma TR
- Medida: Se detecta el 99,99% de los mensajes desordenados.

4.2.2. Disponibilidad:

Es necesario que los datos capturados no se pierdan, más allá de delays o caída de comunicaciones. Debe estar disponible para tomar la decisión correcta sobre un determinado fenómeno.

Escenario 1:

Las TR no pueden estar más de dos minutos sin responder.

- Fuente: Mensaje de la TR a la EC
- Estimulo: Omisión del mensaje
- Artefacto: Terminal Remota
- Entorno: Normal
- Respuesta: Pasar a modo triangulación con TRs cercanas
- Medida: La TR es virtualizada a lo sumo 2 minutos después de detectarse inactiva

Escenario 2:

Si la triangulación no es posible se debe usar el servicio provisto por BiggestSatellit

- Fuente: Mensaje de la TR a la EC
- Estimulo: Caída de una de las TR involucradas en la triangulación
- Artefacto: Modulo de virtualización
- Entorno: Normal
- Respuesta: Se procede a consumir el servicio de BiggestSatellit
- Medida: La TR es reemplazada por la información de BiggestSatellite en a lo sumo 1 minuto

Escenario 3:

Se debe detectar la recuperación de una TR inmediatamente para evitar el gasto en

servicios de terceros.

- Fuente: Interno
- Estimulo: La TR responde diciendo que esta activa
- Artefacto: Modulo de virtualización
- Entorno: Normal
- Respuesta: Se suspende la virtualización de la TR
- Medida: Se comienza a utilizar la TR real en no mas de 10 segundos de detectada su recuperación

Escenario 4:

Tolerancia a perdida de datos

- Fuente: TR
- Estimulo: Omisión de mensaje
- Artefacto: Modulo de recepción de datos
- Entorno: Normal
- Respuesta: Se repite el pedido del dato perdido
- Medida: El 99,9% de los datos llega correctamente.

4.2.3. Performance:

Los pronósticos deben ser producidos en tiempos razonables. Una tardanza en los datos podría llevar a la detección tardía de un fenómeno meteorológico y con ello tardanza para tomar las desiciones pertinentes.

Escenario 1:

La Estación Central debe informar en no mas de 5 minutos desde el sensado de la información al Ministerio de Planificación.

- Fuente: Interno
- Estimulo: Actualización del cálculo del modelo
- Artefacto: Motor del modelo matemático
- Entorno: Normal
- Respuesta: El modelo actualiza el estado
- Medida: Los resultados deben ser informados en menos de 5 minutos

Escenario 2:

Se debe asegurar la vigencia de los datos expuestos a la población en un periodo de no mas de 1 hora.

- Fuente: Interno
- Estimulo: Publicación de datos al público
- Artefacto: Módulo
- Entorno: Normal
- Respuesta: Los datos estan disponibles al público general
- Medida: Los datos expuestos deben tener una vigencia de a lo sumo 1 hora desde que fueron sensados.

Escenario 3:

El monitoreo del sistema debe efectuarse en tiempo casi real

- Fuente: Interna
- Estimulo: una TR cambia de estado
- Artefacto: Modulo de monitoreo de red
- Entorno: Normal
- Respuesta: Se modifica la información sobre la TR adecuadamente
- Medida: El nuevo estado debe ser informado en menos de 15 segundos.

4.2.4. Seguridad:

Transmisión de datos segura, evitando intrusos. Debemos estar seguros de que los datos que manejamos corresponden fecientemente de las TRs

Escenario 1:

Los datos no deben alterarse en el camino de las TR a la EC.

- Fuente: Atacante desconocido
- Estimulo: Intento de modificar datos
- Artefacto: Módulo de recepción de datos
- Entorno: Normal
- Respuesta: Se reconoce la modificación y se repite la petición
- Medida: 99,9% de los datos alterados deben ser identificados

Escenario 2:

Se debe verificar que el emisor de los mensajes sea quien dice ser

- Fuente: una TR
- Estimulo: Envio de mensaje con información del emisor
- Artefacto:
- Entorno: Normal
- Respuesta: Se autentica al emisor y se procesan los datos.
- Medida: 99,999% de las veces se aceptan los datos enviados desde una fuente autenticada

4.2.5. Modificabilidad:

El modelo matemático debe ser facilmente adaptable. El usuario debe tener la capacidad de modificarlo simple

Escenario 1:

Se debe poder agregar TRs sin mayores complicaciones.

- Fuente: Administrador del sistema
- Estimulo: Agregar una TR
- Artefacto: una TR
- Entorno: Normal
- Respuesta: El cambio se efectua sin afectar al sistema
- Medida: La TR se puede instalar en 3 dias.

Escenario 2:

Se deben poder agregar o modificar los modelos matemáticos que operan con los datos.

- Fuente: Clientes del servicio
- Estimulo: Nuevo modelo
- Artefacto: Sistema
- Entorno: Normal
- Respuesta: Se agrega un modelo al motor de resolución
- Medida: El modelo se puede agregar en 24 horas.

Escenario 3:

Se debe poder modificar la agenda de los sensores de cada TR desde la Estación Central.

- Fuente: Operario
- Estimulo: Cambio en la agenda de una TR
- Artefacto: Agenda en la TR
- Entorno: Normal
- Respuesta: La agenda se actualiza correctamente
- Medida: La nueva configuración se aplica en a lo sumo 1 minuto.

4.2.5. Posibles cambios y supuestos fundamentales

Si bien se espera que el sistema sea flexible, carece de sentido pensar que el sistema podría ser modificado para cumplir con cualquier nueva funcionalidad que no tenga que ver con lo que hace ahora. Diseñar de esa manera obligaría a crear una arquitectura extremadamente complicada (y costosa en términos de tiempo y dinero) para funcionalidades que probablemente nunca sucedan. Para ajustarnos mejor a los posibles cambios y no perder el foco en el producto actual se realizó una lista de supuestos fundamentales tomados en cuenta para la arquitectura actual y posibles cambios a futuro que fueron tomados en cuenta a la hora de evaluar la flexibilidad del sistema.

4.2.5.1. Supuestos fundamentales

- El API de Google Maps es lo suficientemente estable y su interfaz se mantiene en sus distintas versiones.
- El metodo de comunicacion de las TRs con la EC va a ser por medio de SMSs o medio similar. De cambiar esto, produciría un cambio considerable en la arquitectura del sistema.
- El presupuesto para la compra de infraestructura del sistema no deberia bajar de forma considerable. Esto atentaria contra el sistema de redundancia y conservacion de los datos.
- Los tipos de sensores colocables en las TRs son conocidos y utilizan un protocolo definido.
- La cantidad de usuarios que utilizaran el sistema web no excede el orden de 1000 requests por segundo.
- Los modelos estan bien definidos y no tienen errores de sintaxis.
- Los expertos tienen vasto conocimiento en el tema.

4.2.5.2. Posibles cambios

- Posibilidad de agregar otras fuentes de informacion como el sistema eolico.
- Posibilidad de nuevos usos a la informacion mostrada en Google Maps.
- Posibilidad de reemplazar a BiggestSatelit por empresa similar.
- Posibilidad de cambiar el tipo de dispatcher de alertas.
- Posibilidad de cambiar la forma de verificacion de disponibilidad de TRs.

4.3. Plan de Riesgos

A continuación listamos los cinco principales riesgos identificados en el proyecto, junto a la clasificación dada a cada uno de ellos en la etapa de análisis de los mismos. Los riesgos se encuentran en orden de prioridad de acuerdo a la clasificación por impacto y probabilidad. Para cada uno de ellos se indica la estrategia de mitigación, el plan de contingencia y el trigger para su ejecución.

4.3.1. Disponibilidad del servicio de SMS

- ID: 1
- Impact: High
- Probability: High
- Priority: 1
- Mitigation Strategy: Implementar un metodo de "Acknowledgment" ante el envio de cualquier mensaje desde las TR a la EC. En caso de no recibir el ACK correspondiente, el paquete es retransmitido.
- Contingency Plan: Utilizar los datos censados con anterioridad ante la imposibilidad de recibir datos desde las TR.
- Plan Trigger: Tests de disponibilidad (monitor de estado de los servicios implementado tempranamente).
- Comments: El alto impacto de este riesgo se debe a la alta prioridad otorgada al atributo de calidad de disponibilidad; asimismo, la falta de calidad de los proveedores de SMS aumentan la probabilidad.

4.3.2. Consumo de energia de las TR

- ID: 2
- Impact: High
- Probability: Low
- Priority: 2
- Mitigation Strategy: Implentar un metodo eficiente de intercambio de modo de energia (panel solar o bateria).
- Contingency Plan: Ante la perdida de energia total del TR, el sistema detectara a la misma como no disponible, tomando los datos de una TR virtual o de BiggestSatelit.
- Plan Trigger: Test de disponibilidad de las TR.

- Comments: Al ser altamente necesaria la fidelidad de los datos enviados, la detección de este problema es muy importante para no generar conclusiones erróneas al ejecutar el modelo matemático.

4.3.3. No disponibilidad de TR

- ID: 3
- Impact: High
- Probability: Med
- Priority: 3
- Mitigation Strategy: Para mitigar esto, se tiene que disponer de mecanismos que detecten la no disponibilidad de forma rápida y eficiente, para proceder a la reparación de las mismas.
- Contingency Plan: Similar al plan de contingencia para el consumo de energía, se va a generar una TR virtual con las más cercanas o se usarán los datos de BiggestSatelit.
- Plan Trigger: Test de disponibilidad de las TR.
- Comments: Al ser altamente necesaria la fidelidad de los datos enviados, la detección de este problema es muy importante para no generar conclusiones erróneas al ejecutar el modelo matemático.

4.3.4. Datos erróneos a partir de los modelos predictivos

- ID: 4
- Impact: High
- Probability: Low
- Priority: 4
- Mitigation Strategy: Revisión de los datos censados y luego enviados a la EC mediante datos históricos y experiencia de los meteorólogos.
- Contingency Plan: Rollback al último modelo que funcionó correctamente en caso que el error provenga del modelo y no de los datos censados.
- Plan Trigger: Revisión de los datos y supervisión de expertos.
- Comments: Este problema es de gran impacto, pero la probabilidad es baja ya que los modelos antes de ser ingresados son detalladamente estudiados por los expertos.

4.3.5. Caída del servicio web para alertar a la población

- ID: 5
- Impact: Low
- Probability: Low
- Priority: 5
- Mitigation Strategy: Redundancia de los servidores web.
- Contingency Plan: En caso de caída del servicio web y un eventual alerta para la población, se van a utilizar otros medios ya considerados como avisos por la radio, televisión o algún otro medio de comunicación masivo.
- Plan Trigger: Test de disponibilidad del Servicio Web.
- Comments: En este caso, este riesgo es de bajo impacto y baja probabilidad, ya que es difícil que el servidor web se caiga ya que está planeado para soportar miles de visitas simultáneas. El bajo impacto se debe a que, en caso de haber en ese momento una emergencia meteorológica, el estado ya cuenta con mecanismos de

alerta. La pagina web es solamente un metodo de seguimiento por parte de la poblacion.

4.3.6. Stoplight Chart:

Este es el spotlight chart parcial con el resultado de la primera iteracion, teniendo los hitos alcanzados y los resultados de las medidas asignadas para los triggers de los planes de contingencias.

Status	Id	Statement	Comments
Green	1	Se implementara en una futura iteracion la triangulacion de datos con otras TRs.	No se implemento el pedido de datos de BiggestSatelit. Esto se hara en la segunda iteracion
Green	2	Se implementara en una futura iteracion la triangulacion de datos con otras TRs	No se implemento el pedido de datos de BiggestSatelit. Esto se hara en la segunda iteracion
Green	3	Se implementara en una futura iteracion la triangulacion de datos con otras TRs. Se implemento uniteracion prototipo de aviso de TR caida en la prueba de concepto.	No se implemento el pedido de datos de BiggestSatelit. Esto se hara en la segunda iteracion
Yellow	4	No se implemento la vuelta atras de los modelos.	Esto se realizara en la segunda iteracion.
Yellow	5	No se implemento el aviso temporal para los usuarios	Esto se realizara en la ultima iteracion.

Documento de Arquitectura

5. Organización del documento

5.1. Partes del documento

Para describir la arquitectura propuesta para el sistema este documento se organiza en las siguientes secciones:

Visión general de la arquitectura: Presentamos una descripción general y de alto nivel de las principales características del producto en el nivel de la arquitectura, con el objetivo de resaltar las decisiones y los fundamentos más importantes detrás del esqueleto general de la misma y la forma en que se orientó su diseño.

Vista híbrida de componentes y conectores de la arquitectura: El propósito de esta vista es delinear detalles mas específicos de la mayoría de los componentes involucrados durante el funcionamiento del sistema. Como lo dice su nombre, el viewtype de esta vista es de "componentes y conectores" mezclado con "deployment" y "modulos, cuyo objetivo es que (junto a toda la información anexa) presenten un panorama completo del diseño arquitectónico.

Arquitectura y atributos de calidad: Aquí repasamos los requerimientos de calidad mas importantes en el sistema, y uno por uno analizamos como y porque la arquitectura propuesta esta en condiciones de responder y satisfacer estas exigencias de calidad.

Plataforma y tecnologías: En las otras secciones que describen la arquitectura tratamos de evitar hacer referencia a tecnologías específicas para poner el foco en otros aspectos de la misma. Es por eso que la idea de esta sección es presentar todas las cuestiones relativas a plataformas y elección de tecnologías particulares que son relevantes a la arquitectura.

5.2. Glosario

En esta sección se presenta una lista de términos y abreviaciones técnicas o de la organización utilizados en el documento:

Término	Abreviatura	Descripción
Terminal Remota	TR	Se refiere a las terminales de medición del estado climático. Cada una cuenta con un conjunto de sensores configurados para enviar datos en un periodo de tiempo determinado.
Estación Central	EC	Es un aglomerado de módulos que se encargan entre otras cosas de recibir los datos de las TRs, procesarlos, almacenarlos e interactuar con otros sistemas.
Secure Socket Layer	SSL	Se refiere a un protocolo que implementa seguridad en un canal de comunicación.

Simple Mail Transfer Protocol	SMTP	Es el protocolo de envío de mail estándar.
Short Message Service	SMS	Servicio de comunicación estándar comunmente usado en telefonía móvil.
Advanced Encryption Standard	AES	Se refiere al estandar de encriptación y cifrado. Sucesor de DES.
Ventana deslizante		Se refiere a un protocolo de transmisión de datos bidireccional de la capa de transporte del modelo OSI.
Secure Hash Algorithm	SHA	Familia de funciones de hash. SHA1 es una de las variantes mas utilizadas.

6. Visión general de la arquitectura

6.1. Consideraciones Generales

Unas de las principales cosas que surgió al analizar los desafíos del problema fue que hay una clara separación entre lo que son fuentes de información, centralización y procesamiento de la misma. Por un lado tenemos Sensores que recolectan datos que serán guardados en la TR para seguridad de los datos. Luego esos mismos datos son llevados a la Estación Central que debe centralizar todos los datos para guardar su registro histórico de métricas meteorológicas. Cabe aclarar que cada dato es procesado y analizado para saber si esta en un intervalo correcto de medición, de esta forma podemos filtrar mediciones erróneas. También utilizamos procesamiento para la parte de cálculo del Modelo Matemático. Esta parte es fundamental que posea un cálculo exacto así como también tener datos correctos para hacer los cálculos. De esta forma, podemos ver estas tres actividades a lo largo de la arquitectura.

Otro aspecto a tener en cuenta es la importancia que tienen los datos en esta Arquitectura. Notemos que el estilo arquitectónico está centrado en los datos meteorológicos y en su accesibilidad. Es por eso que se necesita un alto impacto sobre Bases de Datos y su organización para la correcta sincronización con los diferentes actores.

Un aspecto fundamental de la arquitectura es la comunicación con las TRs. Esta comunicación debe ser segura, como se sabe, pero además debemos tener los suficientes recursos como para poder atender cada mensaje de cada TR de forma satisfactoria. Este es un punto que parece desapercibido pero es importante a la hora de recolectar la información satisfactoriamente.

Para finalizar, debemos darle importancia a la conexión del sistema con el mundo exterior. Hay varios frentes que cubrir. Ya sea a partir de un servidor web siendo la cara visible hacia los ciudadanos. En frente de un cliente para poder actualizar el Modelo Matemático por una aplicación. Una conexión para poder utilizar BiggestSatelit, el Sistema de Cobros y el Sistema Eólico. Y un Operador para tener una consola de operaciones. Todas estas conexiones son importantes ya que el Sistema tiene un alto grado de impacto en la población.

7. Diagrama de arquitectura

7.1. Diagrama

7.2. Detalles del diagrama

Para modelar la arquitectura del sistema se optó por reunir varias vistas en una sola vista híbrida, compuesta por

- Vista de componentes y conectores: se aprecian los componentes (la mayoría de ellos, servidores y módulos) y sus componentes (las conexiones y relaciones entre ellos).
- Vista de módulos: dentro de cada componente se ven los módulos que separan cada tarea realizada por éste. Además se muestran las relaciones de uso entre ellos.
- Vista de deploy: los componentes tienen una clara separación física en el tipo de equipo y ubicación respecto al resto del sistema. También se muestra una breve descripción acerca de los conectores que los relaciona.

Elegimos este enfoque porque nos permitió mostrar en una misma vista tres conceptos fundamentales:

- Separación de responsabilidades: dado un gran volumen de funcionalidades requeridas a través de todo el sistema es necesario poder agrupar qué componentes serán los encargados de cumplir ciertas tareas. La vista de C&C favorece ese aspecto. Pero también los módulos permiten una mayor granularidad sobre éstas.
- Ubicación física: dado que varios componentes se ubicarán a cientos de kilómetros de distancia los unos de los otros, es importante marcar la separación física de cada componente. Y junto con ello, de la responsabilidad de cada tarea. Por otra parte, se debieron aplicar ciertas tácticas y tomar algunas decisiones a nivel de disposición física de los componentes que permitan cumplir con los atributos de calidad referidos a la eficiencia del sistema. Y la mejor manera de reflejarlos es con un diagrama de deploy.
- Interacción entre las partes: podemos considerar una "parte" como un componente. Y, nuevamente, para poder reflejar las decisiones que permitan cumplir con los atributos requeridos de eficiencia (entre otros como la seguridad) hace falta mostrar qué tipo de conexiones existen entre ellos. Además en el presente escenario la comunicación es un punto importante a tener en cuenta por los requerimientos del cliente, donde el más importante es la transmisión de datos de las TRs a la EC mediante SMS.

7.3. Dinámica general del sistema

El sistema consiste de varias terminales remotas ubicadas estratégicamente por el territorio del país. Cada una de ellas posee asociados uno o más sensores los cuales, cada cierto intervalo de tiempo (configurable) o dadas ciertas condiciones, envía un dato reflejando el valor sensado de su magnitud (presión, temperatura, etc). Para ello cada uno utilizará un protocolo sencillo enviando principalmente el dato sensado. El medio físico sobre el cual se comunicará depende del sensor, y puede variar entre conexión por cable o inalámbrica (WiFi o Bluetooth).

La TR irá guardando los datos sensados en un buffer hasta que, pasado cierto tiempo, lo envíe a la estación central. Allí serán recibidos por un servidor encargado de su recolección.

Si llegada la situación se debe triangular una TR virtual o se requiere acceso a BiggestSatelit, será éste servidor (más específicamente, el módulo "Receptor de datos") quien realice los cálculos o consultas correspondientes. También será quien se encargue de la comunicación con las estaciones eólicas.

Luego los datos serán almacenados en una base de datos donde sólo se encuentran aquellos datos sin verificar. Un proceso, la "Tarea de verificación de datos", será el encargado de validarlos cada cierto intervalo de tiempo (pequeño ya que se requiere analizar los datos lo antes posible). La verificación de datos inválidos o outliers se hará en base a configuraciones de ésta tarea y sobre la base de datos válidos. Al encontrarse uno de éstos, se informará al operador y se lo dejará en la base de datos de no verificados (no se descarta porque puede suceder que el operador quiera ingresarlo como un dato válido aunque estadísticamente no lo parezca).

Los datos validados serán uno de los inputs de la Tarea de procesamiento de modelos. El otro será la lista de modelos cargados en la base de datos correspondiente. La ejecución de cada modelo arrojará un resultado, el cual será almacenado en una base de datos destinada a tal fin. A su vez, cada resultado será evaluado por el Verificador de resultados quien dictaminará si se debe o no dar una alerta. En caso afirmativo, el Despachador de alertas avisará al operador dejando un mensaje en la base de datos de la consola de operación, avisará a los ciudadanos mediante el sitio y enviará mails a una lista predefinida (correspondientes a funcionarios del gobierno que deberán dar el alerta y tomar las medidas que correspondan). El proceso de ejecución corre cada cierto intervalo de tiempo (muy pequeño) y sobre un servidor destinado a tal fin (o una granja de ellos, donde también pueden ser varios procesos ejecutándose paralelamente).

La base de datos de Estadísticas y mensajes está destinada a ser una fuente de información con todo lo que el operador pueda necesitar. Las estadísticas se refieren a estado de los sensores, de las TRs, cantidad de datos recibidos, accesos a BS, datos obtenidos del sistema eólico y datos no válidos detectados. Los mensajes son, principalmente, las alertas en caso de detectar un resultado alarmante. El operador podrá acceder a ésta información mediante la aplicación web "Consola de operaciones". Desde allí también podrá dar de alta nuevos modelos y agregar datos inválidos como válidos.

Los ciudadanos podrán acceder a los resultados de los modelos mediante una aplicación web. Dada la criticidad del acceso a la base de datos donde se encuentran por parte del proceso de ejecución, y para agregar mayor seguridad, se decidió no realizar las consultas directamente sobre ésta. En cambio, se mantendrán en un servidor de bases de datos secundario y serán actualizados por la "Tarea de actualización de resultados primarios" que correrá cada cierto intervalo de tiempo (correspondiente a varios minutos ya que estos resultados no se requieren mostrar en tiempo real). De ésta forma se limitan los accesos a la BD primaria, y sólo para los datos que cambiaron desde la última actualización.

A diferencia de los resultados, las alertas que reciben los ciudadanos mediante el sitio deben actualizarse en tiempo real. Es por eso que el Despachador de alertas al detectar alguna inmediatamente la agrega a la base de datos de las aplicaciones web. Así, los usuarios observarán al instante las advertencias. No sólo puede ser un medio para comunicar mensajes de alerta si no que en un futuro podría utilizarse como un boletín de noticias.

Los clientes que contraten el servicio, podrán agregar sus propios modelos y obtener los resultados de los mismos. Para ello accederán a otra aplicación web en donde se autenticarán y podrán hacer altas, bajas y modificaciones de éstos y acceder a la dicha información. Debido a la criticidad de los procesos de ejecución de modelos primarios, no se

utiliza su servidor para la ejecución de los modelos de los clientes. En cambio, se usa el mismo utilizado para la actualización de los resultados primarios en la base secundaria.

De ésta forma se obtiene el circuito completo de recoleccion, almacenamiento, procesamiento, análisis y distribución de la información. De ahora en más se describirán en detalle todas las decisiones de arquitectura realizadas en cada área del sistema.

8. Arquitectura y atributos de calidad

En la siguiente seccion se explica como la arquitectura delineada pretende cumplir con los atributos de calidad requeridos para el producto. Para cumplir con ellos se realizo un analisis detallado de las distintas tacticas y good practices disponibles.

8.1. Seguridad

- Para garantizar la seguridad del sistema, tanto del punto de vista de la confidencialidad como de la integridad de los datos en el envio de las TRs a la EC y viceversa se aplica un metodo de encriptacion AES con un cheksum SHA-1 a cada paquete enviado. La clave utilizada por cada TR es unica para cada una y esta harcodeada en cada appliance. La EC tiene una tabla con las claves de cada de las TRs, la cual es utilizada para encriptar correctamente cada uno de los mensajes.
- Del punto de vista de la seguridad en la pagina web, se siguieron las recomendaciones de la fundacion OWASP (Open Web Applications Security Project), las cuales ofrecen una guia de medidas a tomar ante cada uno de los posibles ataques. Para la autenticacion en el sitio por parte de los clientes se encripta la clave del usuario con AES en la base de datos para la posterior comparacion.

8.2. Disponibilidad

- El primer punto a atacar para el atributo de disponibilidad es uno de los mas importantes del sistema, que consiste en la disponibilidad de las TRs. Segun reuniones con los stakeholders, se desprende que se debe detectar si una TR esta caida en un lapso no mayor a los 2 minutos. Esto fue atacado con la tactica de heartbeat. Cada TR envia mensajes con los datos recolectados a la EC. Este mensaje se toma como un mensaje de disponibilidad haciendo una especie de piggybacking introduciendo dicho mensaje en el original con los datos. Si por alguna razon no llega a enviar ningun mensaje de este estilo, se envia un mensaje a la EC diciendo que esta viva.
- Para asegurar el orden y la recepcion de los datos de las TRs a la EC se va a implementar un sistema de ventana deslizante con un tamaño de un paquete. La EC espera mensajes de cada TR en un tiempo estipulado. Si ese mensaje no se recibio, este manda un NACK a la TR diciendo que no recibio ningun mensaje. Esto puede pasar porque la TR preparo el mensaje pero no se envio o porque se perdio en la red SMS. Esto garantiza el envio ordenado de los paquetes, a la vez que se puede tener una metrica de cuando bien responde cada TR, pudiendo ajustar o mejorar las cosas q fallan, como por ejemplo la disponibilidad del servicio SMS.

- Para no que no exista una perdida de los datos enviados de los sensores a las TRs y de estas a la EC, existen en cada uno de estos dispositivos un buffer que contiene los ultimos mensajes mandados para una futura retransmision o auditoria.
- En el caso de la disponibilidad del servicio web, se decidio establecer un sistema de redundancia entre los servidores para evitar perdida del front-end a los ciudadanos y ante los clientes.
- Cuando un sensor deja de reportar, la TR toma nota de esto y envia un mensaje a la EC avisando al operador que el sensor X en la TR Y esta fallando, para un futuro arreglo.
- En el caso que se corte la comunicacion tanto con BiggestSatellit, con el Sistema de Pagos del Ministerio o con el futuro sistema eólico, se va a reportar como alarma. Se supone, aunque queda por confirmar, que cada uno de estos sistemas tiene un metodo de respaldo para recibir y enviar la informacion que nuestro sistema necesita.

8.3. Performance

- Para garantizar que los datos sensados lleguen en a lo sumo 5 minutos a los responsables en el Ministerio, se realizaron varias pruebas y se determino que el principal cuello de botella del flujo de los datos es el sistema de SMS. Esto esta contemplado y hablado con los stakeholders. Ellos saben que para que esto deje de ser un tema determinante se tendria que invertir una importante suma de dinero, hecho que no va a suceder en el corto plazo. Para acelerar el resto de los procesos al maximo posible, se determino instalar servidores de ultima generacion y conectarlos mediante redes de alta performance, confiables y seguras.
- Para asegurar la vigencia de los datos de la ultima hora a los usuarios, estos se almacenan en una base de datos complementaria a la principal. Esta segunda base de datos tiene como uso exclusivo el almacenamiento de los ultimos resultados de los modelos del Ministerio, de los resultados de los modelos de los clientes, de los mencionados modelos y los datos de los clientes para la autentificacion.
- El monitoreo en tiempo casi real se logra con el constante intercambio de SMSs entre las TRs y la EC. Esto permite tener un estado general casi instantaneo de toda la red telemetrica. Debido a esto y al atributo de disponibilidad, se enforzo el desarrollo del metodo de heartbeat entre las TRs y la EC. Tambien se incorporan datos sobre los sensores y el aviso correspondiente al administrador.

8.4. Confiabilidad

- La confiabilidad del sistema es determinante para el exito del proyecto. Es por eso que siempre se verifican todos los datos, tanto los recibidos de las TRs y otros mecanismos (BiggestSatelit y Sistema Eolico) utilizando datos historicos que establecen una media y un desvio estandar, como tambien los modelos matematicos ingresados al sistema, los cuales son monitoreados dependiendo los resultados que arrojan a partir de los datos verificados. Dichas verificaciones se efectuan en el Sistema de Verificacion de Datos y en el Sistema Primario de

Procesamiento. Ante cualquier anomalia detectada en cualquiera de estos sistemas, se reporta al administrador del sistema. En el caso de los modelos, se puede volver al sistema anterior que funciono correctamente. En el caso de los datos, se descartan para el procesamiento pero quedan guardados para futuras mediciones.

8.5. Modificabilidad

- Uno de los primeros atributos de modificabilidad que se atacaron fue el de la configuracion de los intervalos de envio de datos desde los sensores a las TRs. Estos se configuran desde la EC, enviando un mensaje a la TR correspondiente. Esta reenvia el mensaje al sensor ajustando el tiempo indicado.
- Para agregar una TR se decidio el siguiente metodo: Un operador del Ministerio se dirige a la locacion donde quiere empezar a realizar las mediciones. Una vez colocada junto con los sensores, se le establece una clave AES y un numero de identificacion. Luego, el operador le envia al administrador estos dos ultimos datos para el ingreso en la base de datos de TRs. Una vez listo esto, la EC ya esta lista para recibir los datos de la nueva TR.
- El agregado y modificacion de los modelos se realiza por diferentes medios. Los modelos del Ministerio se agregan a mano mediante el administrador del sistema. En cambio, los modelos de los clientes se agregan por medio del panel de control asignado para tal fin. La modificacion se realiza en forma similar. Los modelos de los clientes se alojan en la base de datos correspondiente en el Servidor secundario de Bases de Datos y los modelos del ministerio se almacenan en el Servidor Primario de Bases de Datos.
- El ingreso de nuevos clientes al sistema lo realiza el administrador de la EC luego de efectuados los pagos correspondientes. Esto se realiza asi ya que al no suponerse un ingreso agresivo de nuevos clientes al sistema, esto no generaria una carga excesiva para el administrador. Si esto ultimo llegase a suceder, se podria revisar la arquitectura para agregar un modulo de ingreso de clientes en el panel de control web ubicado en el Servidor Web.

Seguimiento y avance

12. Informe de seguimiento y avance

Realizamos el siguiente documento para analizar el seguimiento de las tareas realizadas en la primer iteración. Nos basamos fuertemente en el plan de proyecto en el cual se definieron y estimaron las tareas a realizar en esta primera iteración. A partir de ese documento se asignaron los recursos a las tareas de forma conveniente para poder lograr un buen aprovechamiento de los mismos en el tiempo establecido. Notamos que fue buena idea designar todos los recursos, o sea designar tareas que involucren a todo el equipo, sobre tareas que eran importantes para el desarrollo del sistema. De esta forma, a cada uno de los integrantes del equipo de trabajo quedaba bien enterado de como se realizo una determinada tarea que luego iba a impactar al dividir el grupo en dos tareas distintas.

Para poder alcanzar los objetivos de esta primera iteración tratamos de cumplir con la planificación estimada. El seguimiento consistió en actualizar el estado de las tareas a medida que se iban completando. De esta forma todos los integrantes estaban al tanto del grado de avance del proyecto, así como también de lo que falta hacer. La idea es que en todo momento se tenga una noción sobre lo que esta hecho y lo que queda pendiente.

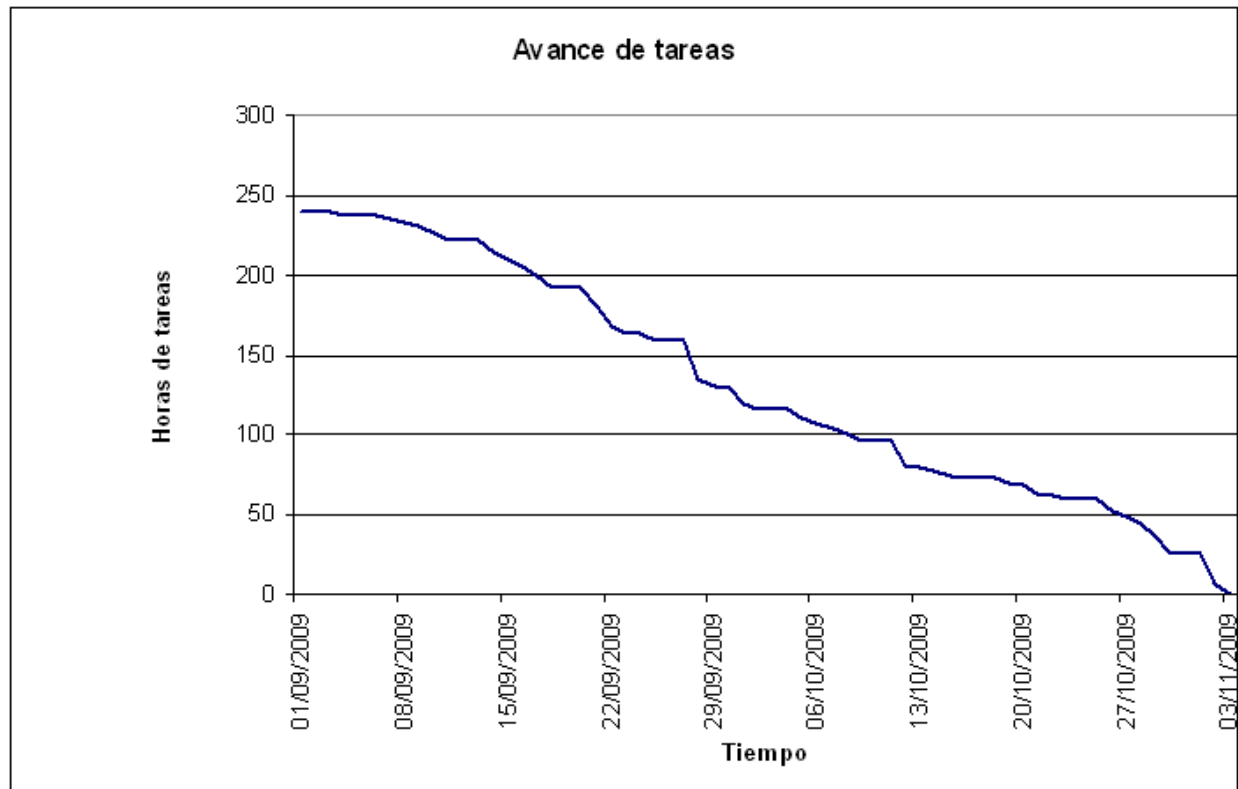
Vamos a definir el avance mas en concreto:

Al momento de entrega de este documento el proyecto posee el siguiente avance.

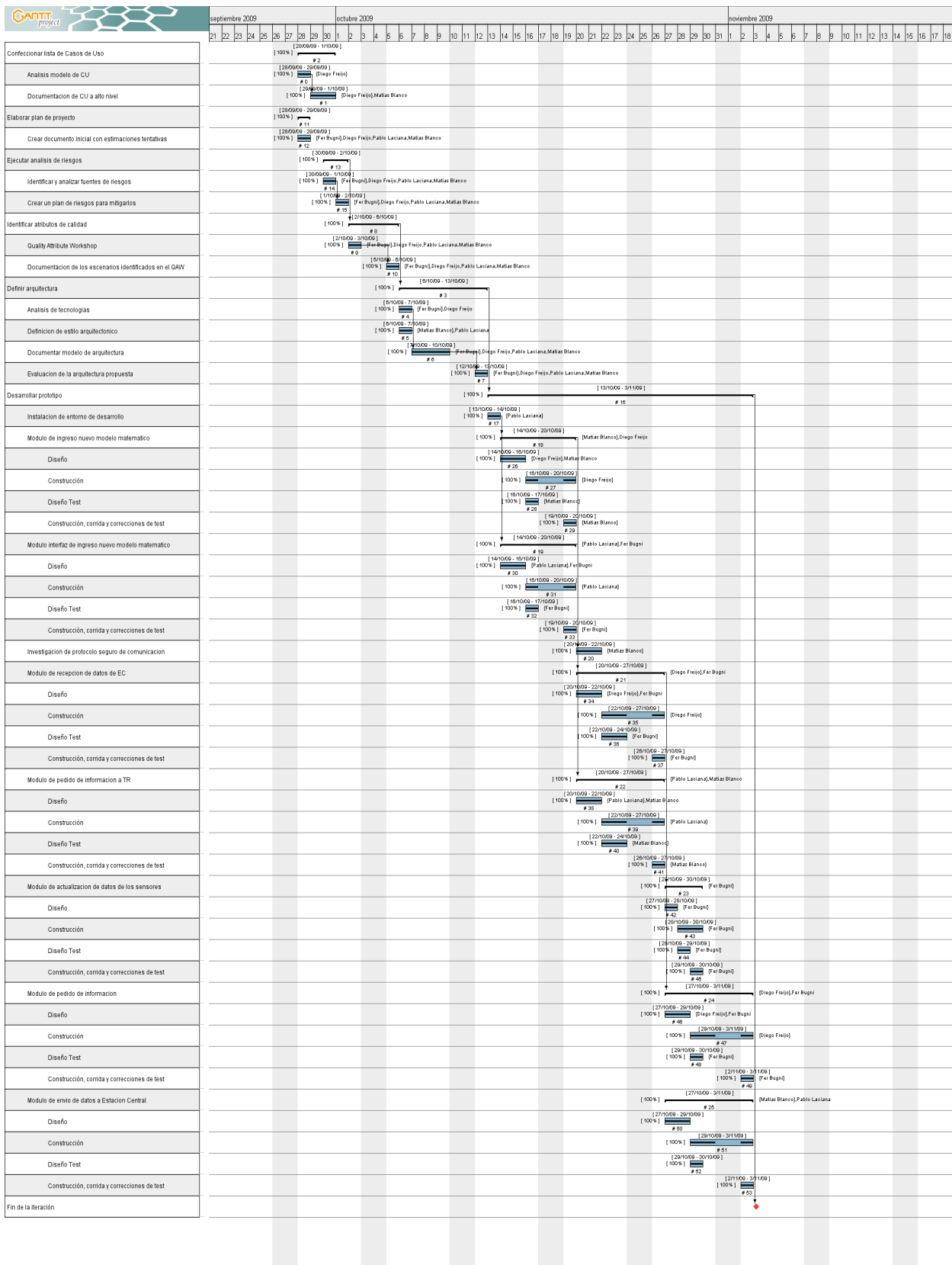
- Se han identificado los Casos de Uso.
- Se han definido los atributos de calidad que debe poseer el sistema asi como sus escenarios principales
- Se han estimado las fechas de los hitos y entregables más importantes que acompañarán el proyecto.
- Se ha definido la arquitectura del proyecto, detallando cuales con los componentes y conectores que componen el sistema y cuales son las tácticas utilizadas para satisfacer los atributos de calidad del sistema. Esta tarea requirió una dedicación sobresaliente de todo el equipo en consideración con el resto de las tareas. El equipo entero debió reunirse varias veces para llegar a una definición de arquitectura completa. Luego la documentación de la misma se delegó a un integrante del mismo.
- Se han definido las fases del proyecto y cuales son las iteraciones de cada una de las fases.
- Hemos especificado qué casos de uso es llevado a cabo en cada una de estas iteraciones.
- Hemos definido qué tareas deben llevarse a cabo en la primera iteración de la fase de elaboración, hemos definido cuánto insumirá cada una de esas tareas y hemos especificado cuales son las precedencias entre cada una de esas tareas.
- Hemos elaborado un Plan de Riesgos para el proyecto.
- Hemos implementado la prueba de concepto sobre una porción del sistema. Una vez elegida la tecnología para realizarla, un solo miembro del equipo pudo realizarla.

A pesar de faltar unos dias para terminar con la primera iteración se logró llegar con los entregables de esta etapa. Esto quiere decir que estimamos con una idea pesimísta pero lo bueno que llegamos a la entrega de los documentos satisfactoriamente.

Para tener un control estricto del avance se decidió que los miembros del equipo reportaran las horas de trabajo. Dado que nuestra estimación original estaba dada en días por tareas y que es mas natural reportar horas, vamos a considerar que un día laboral consta de 4 horas. Considerando nuestra estimación inicial entonces, al inicio de la iteración ninguna tarea había sido realizada. Con el correr del tiempo las horas estimadas van siendo consumidas para finalizar en la fecha de fin de iteración con todas las tareas realizadas. El siguiente gráfico muestra el avance propiamente dicho medido en estos términos.



Para el seguimineto diario utilizamos el ya mencionado Gantt pero le agregamos el porcentaje de cada tarea realizada. Este es el mismo al finalizar la primera iteración.



13. Prueba de concepto

13.1. Descripción

La prueba de concepto consiste en un programa realizado con el framework .NET 3.5, utilizando WinForms. Esto se realizó debido a la facilidad de esta tecnología para generar aplicaciones interactivas.

El archivo POC.exe es el ejecutable para la utilización de esta prueba. Una vez iniciado, se visualizan 3 grillas y un log. En la grilla superior, se indican las TRs con las cuales el sistema va a interactuar. Estas son generadas de manera aleatoria al apretar el botón "Generar TRs". Esto se realizó para no depender de un número fijo de ellas, y mostrar la escalabilidad del sistema. Una vez generadas, se puede manipular algunas de las variables de las mismas, como romper algún sensor (acción que "rompe" aleatoriamente uno o varios sensores de una o varias TRs) o bajar la conectividad, es decir, reducir virtualmente la disponibilidad del servicio SMS.

Una vez realizado esto, al apretar el botón "Enviar Sensor a TR". Esto enviará los datos de cada sensor sano y activado a su TR correspondiente. Una vez hecho esto, se va a poder ver en la segunda columna de la primera grilla como la TR almacena los datos enviados. Siempre se mantienen los datos de la última recepción del sensor.

Cuando cada TR tiene los datos de sus sensores, al apretar el botón "Enviar TR a EC", los datos se preparan para enviar a la EC. Cada TR genera un paquete con su nombre y la información recolectada. Con este paquete, se genera un SMS. Para simular la pérdida de SMSs, se utiliza la condición del servicio actual multiplicado por un número aleatorio entre 0 y 100. Este producto se divide por 2 para tener un número entre 1 y 100. Si el número es mayor a 50, el mensaje se envía exitosamente, en caso contrario, el mensaje se pierde. La EC, luego de hacer el pedido, espera un tiempo. Pasado ese tiempo envía un aviso de no recepción a la TR para que reenvíe el paquete. Este mensaje se envía a través de un SMS, el cual sigue la misma política de aleatoriedad para el envío. La TR siempre guarda el último mensaje que envió y al recibir el aviso de no recepción, envía nuevamente el SMS. Si la EC no recibió el mensaje en 5 intentos, escribe en el log que la TR parece estar caída, ya que no puedo establecer la comunicación.

Cada mensaje que se recibió satisfactoriamente se muestra en la segunda grilla. Las TRs no envían en orden según su ID, sino que eso es aleatorio. En el log de la derecha se pueden ver mensajes de pedido de información, recepción de la misma, envío de aviso de no recepción, avisos de TR caída y fin de la recepción.

Por último, en la grilla inferior se muestran los modelos que se encuentran ingresados en el sistema. Para ingresarlos, se aprieta en el botón "Nuevo Modelo". Esto genera la muestra de un formulario donde se ingresa el nombre del modelo y la fórmula. Arriba del campo donde se ingresa la fórmula está explicado que nombre se usa para cada variable, que luego va a ser reemplazado por los datos sensados.

13.2. Tecnologías utilizadas

Para ejecutar esta prueba, necesita disponer de un sistema con el sistema operativo Windows con el framework .NET 3.5 instalado.

